DETAILED ACTION

Remarks

1. Receipt of Applicant's Amendment, filed on 05/19/2010, is acknowledged. The amendment includes the cancellation of Claims 1-9, 11-13, 17-19, and 22-26, and the amending of Claims 10, 20, 27, and 30-32.

Claim Objections

2. The objections raised in the office action mailed on 02/19/2010 have been overcome by applicant's amendments received on 05/19/2010.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

5. Claims 10, 14-16, 20-21, and 27-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Win et al.** (U.S. Patent 6,453,353) in view of **Faybishenko et al.** (U.S. PGPUB 2003/0158839), and further in view of **Driesch et al.** (U.S. PGPUB 2003/0065648)

6. Regarding claim 10, **Win** teaches a method comprising:

A) assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
B) wherein the assigned metadata requirements specify conditions required for successful execution of the functional module (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
C) wherein at least one condition defines at least one user role required for successful; execution of the functional module (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
H) obtaining a list of functional modules that are accessible from within the application used during the query session (Abstract, Column 6, lines 10-16, lines 41-65);
I) identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements (Abstract, Column 6, lines 10-16, lines 41-65); and
J) providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute (Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **"assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database"** as "Roles determine what resources a User can access. Further, each role may require a set of information that is available in resources" (Column 5, lines 44-46) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"wherein the assigned metadata requirements specify conditions required for successful execution of the functional module"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the

Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"wherein at least one condition defines at least one user role required for successful; execution of the functional module"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"obtaining a list of functional modules that are accessible from within the application used during the query session"** as "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15).

Win does not explicitly teach:

D) collecting runtime metadata relating to one or more result fields in a query statement;

E) wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement;

F) wherein the runtime metadata is collected after composition of the query statement.

Faybishenko, however, teaches **"collecting runtime metadata relating to one or more result fields of a query"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112), **"wherein the metadata is collected after composition of the query"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112), and **"wherein the runtime metadata is collected after composition of the query statement"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits,

searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Win and **Faybishenko** do not explicitly teach:

G) wherein the runtime metadata is collected before the query statement is submitted for execution.

Driesch, however, teaches **"wherein the runtime metadata is collected before the query statement is submitted for execution"** as "In addition, steps are taken to determine whether or not log information for the various queries will be collected. Specifically, the user-supplied predictive query threshold is utilized and compared to one or more cost factors (i.e., factors used in determining the optimal execution plan). One illustrative cost factor is an estimated runtime for a particular query. In one embodiment, the determination of whether or not to collect log information comprises determining whether to initiate a monitor process for a particular query at all. In this case, the determination is made prior to execution of the selected execution plan" (Paragraph 33), "Step 206 represents generating a compiled set of runtime structures, called an "access plan" or "execution plan", from the compiled SQL statements. The

access plan represents the computer-generated sequence of operations to obtain the data specified by the user query. Generation of the access plan involves consideration of both the available access paths (indexes, sequential reads, etc.) and system held statistics on the data to be accessed (the size of the table, the number of distinct values in a particular column, etc.), to choose what the RDBMS processor considers to be the most efficient access plan for the query" (Paragraph 34), and "At step 306, the query is marked for monitoring. That is, some indication that the query will be monitored is made. In one embodiment, for example, a flag may be set for the query. At step 308, the log 134 is initialized. The plan information generated at step 206 (described above with reference to FIG. 2) is then written to the log 134 at step 310. Processing then proceeds to step 312. At step 312, a query cursor for the selected plan is opened in preparation to execute the plan. A loop is then entered at step 314 for each record for the cursor. At a first step of the loop, represented by step 316, a record for the cursor is fetched. At step 318, the method 210 determines whether the query for which the record is fetched is being monitored. This determination is made according to the implementation of step 306 described above. For example, if a flag value was set, then a flag value is checked at step 318. If the query is not being monitored, the method 210 returns to step 314 to begin processing for the next record. If, however, the query is being monitored, then the I/O activity for the query is logged at step 320. Processing then returns to step 314. Once all of the records for the cursor have been fetched (i.e., the query has been executed), the method 210 exits at step 322" (Paragraphs 36-37).

The examiner notes that "plan information" includes data about the tables being accessed to retrieve desired data form a SQL query. Moreover, the plan information is logged (See "The plan information generated at step 206 (described above with reference to FIG. 2) is then written to the log 134 at step 310") before the query is executed (See that "At step 312, a query cursor for the selected plan is opened in preparation to execute the plan").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Driesch's** would have allowed **Win's** and **Faybishenko's** to provide a method to

reduce overhead when logging information about queries, as noted by **Driesch** (Paragraph 10).

Regarding claim 14, **Win** further teaches a method comprising:
A) wherein obtaining metadata associated with the functional module comprises examining a signature validation (Column 6, lines 1-3, Column 14, lines 34-43).

The examiner notes that **Win** teaches **"wherein obtaining metadata associated with the functional module comprises examining a signature validation"** as "users may log in either with a digital certificate or by opening a login page URL with a web browser and entering a name and password" (Column 6, lines 1-3).

Regarding claim 15, **Win** further teaches a method comprising:
A) wherein the metadata associated with at least one of the functional modules comprises at least one of: one or more input parameters required for successful execution of the functional module, one or more output parameters required for successful execution of the functional module, and a security credential required to execute the functional module (Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **"wherein the metadata associated with at least one of the functional modules comprises at least one of one or more input parameters required for successful execution of the functional module; one or more output parameters required for successful execution of the functional module; and a credential of a user authorized to execute the functional module"** as "The Authentication Client Module and Access Menu Module authenticates a user by verifying the name and password with the Registry Server 108" (Column 6, lines 42-44).

Regarding claim 16, **Win** does not explicitly teach a method comprising:
A) wherein at least one of the functional modules analyzes query results.

Faybishenko, however, teaches **"wherein at least one of the functional modules analyzes query results"** as "In one embodiment a QRP adapter may monitor

or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that storing the click history of users teaches the claimed analyzing of query results.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 20, **Win** teaches computer readable storage medium comprising:

A) assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);

B) wherein the assigned metadata requirements specify conditions required for successful execution of the functional module (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);

C) wherein at least one condition defines at least one user role required for successful execution of the functional module (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);

H) obtaining a list of functional modules accessible from within the application (Abstract, Column 6, lines 10-16, lines 41-65);
I) identifying a limited subset of the functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements (Abstract, Column 6, lines 10-16, lines 41-65); and
J) providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute (Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **"assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database"** as "Roles determine what resources a User can access. Further, each role may require a set of information that is available in resources" (Column 5, lines 44-46) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"wherein the assigned metadata requirements specify conditions required for successful execution of the functional module"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"wherein at least one condition defines at least one user role required for successful; execution of the functional module"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources"

(Column 6, lines 13-14). The examiner further notes that **Win** teaches **"obtaining a list of functional modules accessible from within the application"** as "a list of authorized resources" (Column 6, lines 13-14) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"identifying a limited subset of the functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15).

Win does not explicitly teach:

D) collecting runtime metadata relating to one or more result fields in a query statement;

E) wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement;

F) wherein the runtime metadata is collected after composition of the query statement.

Faybishenko, however, teaches **"collecting runtime metadata relating to one or more result fields in a query statement"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally

the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112), **"wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112), and **"wherein the runtime metadata is collected after composition of the query statement"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a

registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Win and **Faybishenko** do not explicitly teach:

G) wherein the runtime metadata is collected before the query statement is submitted for execution.

Driesch, however, teaches **"wherein the runtime metadata is collected before the query statement is submitted for execution"** as "In addition, steps are taken to determine whether or not log information for the various queries will be collected. Specifically, the user-supplied predictive query threshold is utilized and compared to one or more cost factors (i.e., factors used in determining the optimal execution plan). One illustrative cost factor is an estimated runtime for a particular query. In one embodiment, the determination of whether or not to collect log information comprises determining whether to initiate a monitor process for a particular query at all. In this case, the determination is made prior to execution of the selected execution plan" (Paragraph 33), "Step 206 represents generating a compiled set of runtime structures, called an "access plan" or "execution plan", from the compiled SQL statements. The access plan represents the computer-generated sequence of operations to obtain the data specified by the user query. Generation of the access plan involves consideration of both the available access paths (indexes, sequential reads, etc.) and system held statistics on the data to be accessed (the size of the table, the number of distinct values in a particular column, etc.), to choose what the RDBMS processor considers to be the most efficient access plan for the query" (Paragraph 34), and "At step 306, the query is marked for monitoring. That is, some indication that the query will be monitored is

made. In one embodiment, for example, a flag may be set for the query. At step 308, the log 134 is initialized. The plan information generated at step 206 (described above with reference to FIG. 2) is then written to the log 134 at step 310. Processing then proceeds to step 312. At step 312, a query cursor for the selected plan is opened in preparation to execute the plan. A loop is then entered at step 314 for each record for the cursor. At a first step of the loop, represented by step 316, a record for the cursor is fetched. At step 318, the method 210 determines whether the query for which the record is fetched is being monitored. This determination is made according to the implementation of step 306 described above. For example, if a flag value was set, then a flag value is checked at step 318. If the query is not being monitored, the method 210 returns to step 314 to begin processing for the next record. If, however, the query is being monitored, then the I/O activity for the query is logged at step 320. Processing then returns to step 314. Once all of the records for the cursor have been fetched (i.e., the query has been executed), the method 210 exits at step 322" (Paragraphs 36-37).

The examiner notes that "plan information" includes data about the tables being accessed to retrieve desired data form a SQL query. Moreover, the plan information is logged (See "The plan information generated at step 206 (described above with reference to FIG. 2) is then written to the log 134 at step 310") before the query is executed (See that "At step 312, a query cursor for the selected plan is opened in preparation to execute the plan").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Driesch's** would have allowed **Win's** and **Faybishenko's** to provide a method to reduce overhead when logging information about queries, as noted by **Driesch** (Paragraph 10).

Regarding claim 21, **Win** does not explicitly teach a computer readable storage medium comprising:

A) wherein the application is a query building application.

Faybishenko, however, teaches **"wherein the application is a query building application"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 27, **Win** teaches a data processing system comprising:

A) a data repository (Column 5, lines 13-15);

B) a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules (Abstract, Column 6, lines 10-16, lines 41-65);

C) wherein at least one condition defines at least one user role required for successful execution of the functional modules (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);

D) an application from which the functional modules are accessible (Abstract, Column 6, lines 10-16, lines 41-65);

G) present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and

the metadata requirements associated with the functional modules (Abstract, Column 5, lines 66-67-Column 6, lines 1-16).

The examiner notes that Win teaches **"a data repository"** as "The system 2 enables organizations to register information sources or Resources and register Users of the information in a central repository" (Column 5, lines 13-15). The examiner further notes that **Win** teaches **"a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules"** as "a list of authorized resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"an application from which the functional modules are accessible"** as "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"wherein at least one condition defines at least one user role required for successful execution of the functional modules"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14).

Win does not explicitly teach:

F) <u>wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement</u>;

Faybishenko, however, teaches **"wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Win and **Faybishenko** do not explicitly teach:

E) wherein the application is configured to: after composition of a query statement, but before the query statement is submitted for execution, collect runtime metadata related to one or more result fields in the query statement.

Driesch, however, teaches **"wherein the application is configured to: after composition of a query statement, but before the query statement is submitted for execution, collect runtime metadata related to one or more result fields in the query statement"** as "In addition, steps are taken to determine whether or not log information for the various queries will be collected. Specifically, the user-supplied predictive query threshold is utilized and compared to one or more cost factors (i.e.,

factors used in determining the optimal execution plan). One illustrative cost factor is an estimated runtime for a particular query. In one embodiment, the determination of whether or not to collect log information comprises determining whether to initiate a monitor process for a particular query at all. In this case, the determination is made prior to execution of the selected execution plan" (Paragraph 33), "Step 206 represents generating a compiled set of runtime structures, called an "access plan" or "execution plan", from the compiled SQL statements. The access plan represents the computer-generated sequence of operations to obtain the data specified by the user query. Generation of the access plan involves consideration of both the available access paths (indexes, sequential reads, etc.) and system held statistics on the data to be accessed (the size of the table, the number of distinct values in a particular column, etc.), to choose what the RDBMS processor considers to be the most efficient access plan for the query" (Paragraph 34), and "At step 306, the query is marked for monitoring. That is, some indication that the query will be monitored is made. In one embodiment, for example, a flag may be set for the query. At step 308, the log 134 is initialized. The plan information generated at step 206 (described above with reference to FIG. 2) is then written to the log 134 at step 310. Processing then proceeds to step 312. At step 312, a query cursor for the selected plan is opened in preparation to execute the plan. A loop is then entered at step 314 for each record for the cursor. At a first step of the loop, represented by step 316, a record for the cursor is fetched. At step 318, the method 210 determines whether the query for which the record is fetched is being monitored. This determination is made according to the implementation of step 306 described above. For example, if a flag value was set, then a flag value is checked at step 318. If the query is not being monitored, the method 210 returns to step 314 to begin processing for the next record. If, however, the query is being monitored, then the I/O activity for the query is logged at step 320. Processing then returns to step 314. Once all of the records for the cursor have been fetched (i.e., the query has been executed), the method 210 exits at step 322" (Paragraphs 36-37).

The examiner notes that "plan information" includes data about the tables being accessed to retrieve desired data form a SQL query. Moreover, the plan information is

logged (See “The plan information generated at step 206 (described above with reference to FIG. 2) is then written to the log 134 at step 310”) before the query is executed (See that "At step 312, a query cursor for the selected plan is opened in preparation to execute the plan").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Driesch’s** would have allowed **Win’s** and **Faybishenko’s** to provide a method to reduce overhead when logging information about queries, as noted by **Driesch** (Paragraph 10).

Regarding claim 28, **Win** does not explicitly teach a data processing system comprising:

A) wherein the data repository comprises XML data structures used to store runtime metadata.

Faybishenko, however, teaches **“wherein the data repository comprises XML data structures used to store runtime metadata”** as “In some embodiments, users and end applications (consumers 140) may present queries to a distributed information discovery network as arbitrary XML. Schema selection may be performed by HTTP header specification, in some embodiments. In one embodiment, queries presented by consumers 140 may adhere to specific queryspaces. In some embodiments, queries may be routed to the appropriate provider 120 by sending requests (e.g. XML requests) over HTTP” (Paragraph 54).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko’s** would have allowed **Win’s** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 29, **Win** further teaches a data processing system comprising:

A) wherein the data repository comprises relational database tables used to store runtime metadata (Column 5, lines 13-15, Column 7, lines 1-6).

The examiner notes that **Win** teaches **"wherein the data repository comprises relational database tables used to store runtime metadata"** as "The Registry Repository is structured as a database. For example, the Registry Repository may be an SQL Server relational database management system, the Oracle 7® database, etc." (Column 7, lines 1-6). The examiner further notes that it is common knowledge that relational databases store data in tables.

Regarding claim 30, **Win** does not explicitly teach a method comprising:
A) wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type.

Faybishenko, however, teaches **"wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control

over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 31, **Win** does not explicitly teach a computer readable storage medium comprising:

A) wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type for the result field.

Faybishenko, however, teaches **"wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type for the result field"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 32, **Win** does not explicitly teach a data processing system comprising:

A) wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type for the result field.

Faybishenko, however, teaches **"wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type for the result field"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Response to Arguments

7. Applicant's arguments with respect to claims 10, 14-16, 20-21, and 27-32 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. PGPUB 2002/0083075 issued to **Brummel et al.** on 27 June 2002. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 6,757,898 issued to **Ilsen et al.** on 29 June 2004. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2003/0140043 issued to **Hotchkiss et al.** on 24 July 2003. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 20040249674 issued to **Eisenberg** on 09 December 2004. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2003/0229623 issued to **Chang et al.** on 11 December 2003. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2002/0091836 issued to **Moetelli** on 11 July 2002. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 6,430,556 issued to **Goldberg et al.** on 06 August 2002. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 7,027,975 issued to **Pazandak et al.** on 11 April 2006. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2005/0080656 issued to **Crow et al.** on 14 April 2005. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2002/0147724 issued to **Fries et al.** on 10 October 2002. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 6,938,035 issued to **Driesch et al.** on 30 August 2005. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

May 28, 2010
/Mahesh H Dwivedi/
Examiner, Art Unit 2168

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168